Reduced complexity in M/Ph/c/N queues

Alexandre Brandwajn, Thomas Begin

Reduced complexity in M/Ph/c/N queues

Alexandre BRANDWAJN, Thomas BEGIN

RESEARCH REPORT

N° 8303

01/05/2013

Project-Team DANTE

ISSN 0249-6399

Reduced complexity in M/Ph/c/N queues

Alexandre Brandwjan [1] , Thomas Begin [2]

Project-Team DANTE

Research Report N° 8303 — 01/05/2013 —15 pages.

Abstract: A large number of real-life systems can be viewed as instances of the classical M/G/c/N queue. The exact analytical solution of this queueing model is not known, and a frequently-used approach is to replace the general service time distribution by a phase-type distribution. The advantage of this approach is that the resulting M/Ph/c/N queue can be described by familiar balance equations. The downside is that the size of the resulting state space suffers from the "dimensionality curse", i.e., exhibits combinatorial growth as the number of servers and/or phases increases.

To circumvent this complexity issue, we propose to use, instead of the classical full state description, a reduced state description in which the state of only one server is represented explicitly, while the other servers are accounted for through their rate of completions. The accuracy of the resulting approximation is generally good and, moreover, tends to improve as the number of servers in the system increases. Its computational complexity in terms of the number of states grows only linearly in the number of servers and phases, thus making the numerical solution of such queues with hundreds of servers and a reasonable number of phases computationally affordable.

Key-words: Multiple servers, general service, finite buffer, M/Ph/c/N queue, reduced-state approximation, linear complexity.

[1] UCSC – alexb@soe.ucsc.edu

[2] LIP – Thomas.begin@ens-lyon.fr

RESEARCH CENTRE
GRENOBLE - RHÔNE-ALPES

Inovallée
655 avenue de l'Europe - Montbonnot
38334 Saint Ismier Cedex France

Complexité réduite pour les files M/Ph/c/N

Résumé : De nombreux systèmes réels peuvent être vus comme des instantiations de la file classique M/G/c/N. La solution analytique exacte de ce modèle file d'attente demeure inconnu, et une approche fréquemment employée consiste à remplacer la distribution générale du temps de service par une distribution de type phase. L'avantage de cette approche est que la file M/Ph/c/N résultante peut être décrite par des équations d'équilibres familières. Le désavantage est que la taille de l'espace d'état résultant souffre du "dimensionality curse", i.e., il croît combinatoirement lorsque le nombre de serveurs et /ou de phases s'accroît.

Pour pallier ce problème de complexité, nous proposons d'utiliser, à la place de la description d'état classique complète, une description d'état réduite dans laquelle l'état d'un seul serveur est représenté explicitement, les autres étant pris en compte par leur taux de fins de service. La précision de l'approximation résultante est généralement bonne et, en plus, elle tend à s'améliorer lorsque le nombre de serveurs dans le système s'accroît. Sa complexité de calcul en terme de nombre d'état s'accroît seulement linéairement avec le nombre de serveurs et de phases, ce qui permet la résolution numérique de ce type de files avec des centaines de serveurs et un nombre raisonnable de phases.

Mots clés : Serveurs multiples, service général, tampon fini, file M/Ph/c/N, approximation par état réduit, complexité linéaire

1. INTRODUCTION

A large number of real-life systems (such as call centers, multi-core processors, etc) can be modeled as instances of the classical *M/G/c/N* queue (i.e. an *M/G/c* queue with a maximum of N requests in the system) if the pattern of request arrivals is relatively well behaved and can be represented by a quasi-Poisson process. The exact analytical solution of this queueing model is not known, and existing approximations are either difficult to evaluate computationally [HOK78, MIY86, cites par Kimura] or fail to capture the potentially important dependence of the performance of such a queueing system on higher-order properties of the service time distribution [GOU96, SMI03]. A critical review of several of these approximations can be found in the papers by Kimura [KIM93, KIM96]. A more recent reference is the work by Smith [SMI03] which proposes an approximation for the loss probability based only on the first two moments of the service time.

Outside simulation, a frequently-used approach is to replace the general service time distribution by a phase-type distribution, as it is known that any distribution can be approximated arbitrarily closely by a distribution of the latter type [JOH88]. The obvious advantage of this approach is that, in steady state, the resulting *M/Ph/c/N* queue can be described by familiar balance equations. Generally speaking, these balance equations can be obtained using one of two possible state descriptions involving the current number of request in the system and a vector to represent the state of the servers. The first one is the vector of the current number of servers in each phase of the service process. The second possible description is the vector of the current phases for each server (note that the servers are assumed to be homogenous but they are not synchronized.) This latter state description is always less thrifty than the first one and rarely, if ever, used. Both descriptions exhibit combinatorial growth as the number of phases and the number of servers grow.

Several methods (e.g, matrix geometric, direct iteration [SEE86, RAM85a, RAM85b, LAT93, LAT94]) can be used to solve these equations numerically. As long as the number of servers and service phases remains small these methods work fine. However, it is also known that the size of the system of equations to be solved suffers from what has been termed the "dimensionality curse", in that the number of states grows combinatorially as the number of servers and phases increase. Thus, for larger numbers of servers, these methods become impractical, and there is a clear need for an approach that would handle larger numbers of servers (say, hundreds) with a reasonable number of service time phases.

Our goal in this paper is to propose a different approach to the approximate evaluation of the *M/Ph/c/N* queue. Our approach is based on a reduced state description to circumvent the explosion of the number of states discussed above. In the following section, we describe in more detail the queueing system considered and we introduce the reduced state description. In Section 3, we present numerical results illustrating the accuracy of the proposed approximation, as well as the savings in the size of the state space. Section 4 concludes this paper.

2. MODEL, STATE DESCRIPTION AND SOLUTION

Consider the *M/Ph/c/N* queue represented in Figure 1. The times between arrivals are assumed to memoryless (quasi-Poisson) and the service times are represented as a phase-type distribution with a total of b phases. There are c homogenous servers in our system and the buffer space is restricted to a maximum of N requests in the systems (queued and in service.) We assume that $N > c$, since otherwise there would be no queue build up possible. We also assume that the rate of arrivals and the parameters of the service process may depend on the current number of requests in the system, denoted by n. This type of state dependence is useful, in particular, to represent arrivals from a finite number of exponential sources and service process which varies with the workload. The detailed notation used in our paper is given in Table 1.

We consider the stationary behavior of such a queue. As mentioned in the introduction, the state of our system could be fully described by the total current number of requests in the system and the numbers of requests in each phase of the service process or, alternatively, by the current total number of requests and the current phase of each server. Instead of such a full state description, we propose to use a reduced state description in which we select one server among the c servers and describe the system by the total number of requests and the current phase of the selected server, (n,i). For $n < c$, with probability $(c-n)/c$ the selected server may be idle, in which case we use the value $i = 0$ to denote its idle state.

1. INTRODUCTION

A large number of real-life systems (such as call centers, multi-core processors, etc) can be modeled as instances of the classical *M/G/c/N* queue (i.e. an *M/G/c* queue with a maximum of N requests in the system) if the pattern of request arrivals is relatively well behaved and can be represented by a quasi-Poisson process. The exact analytical solution of this queueing model is not known, and existing approximations are either difficult to evaluate computationally [HOK78, MIY86, cites par Kimura] or fail to capture the potentially important dependence of the performance of such a queueing system on higher-order properties of the service time distribution [GOU96, SMI03]. A critical review of several of these approximations can be found in the papers by Kimura [KIM93, KIM96]. A more recent reference is the work by Smith [SMI03] which proposes an approximation for the loss probability based only on the first two moments of the service time.

Outside simulation, a frequently-used approach is to replace the general service time distribution by a phase-type distribution, as it is known that any distribution can be approximated arbitrarily closely by a distribution of the latter type [JOH88]. The obvious advantage of this approach is that, in steady state, the resulting *M/Ph/c/N* queue can be described by familiar balance equations. Generally speaking, these balance equations can be obtained using one of two possible state descriptions involving the current number of request in the system and a vector to represent the state of the servers. The first one is the vector of the current number of servers in each phase of the service process. The second possible description is the vector of the current phases for each server (note that the servers are assumed to be homogenous but they are not synchronized.) This latter state description is always less thrifty than the first one and rarely, if ever, used. Both descriptions exhibit combinatorial growth as the number of phases and the number of servers grow.

Several methods (e.g, matrix geometric, direct iteration [SEE86, RAM85a, RAM85b, LAT93, LAT94]) can be used to solve these equations numerically. As long as the number of servers and service phases remains small these methods work fine. However, it is also known that the size of the system of equations to be solved suffers from what has been termed the "dimensionality curse", in that the number of states grows combinatorially as the number of servers and phases increase. Thus, for larger numbers of servers, these methods become impractical, and there is a clear need for an approach that would handle larger numbers of servers (say, hundreds) with a reasonable number of service time phases.

Our goal in this paper is to propose a different approach to the approximate evaluation of the *M/Ph/c/N* queue. Our approach is based on a reduced state description to circumvent the explosion of the number of states discussed above. In the following section, we describe in more detail the queueing system considered and we introduce the reduced state description. In Section 3, we present numerical results illustrating the accuracy of the proposed approximation, as well as the savings in the size of the state space. Section 4 concludes this paper.

2. MODEL, STATE DESCRIPTION AND SOLUTION

Consider the *M/Ph/c/N* queue represented in Figure 1. The times between arrivals are assumed to memoryless (quasi-Poisson) and the service times are represented as a phase-type distribution with a total of b phases. There are c homogenous servers in our system and the buffer space is restricted to a maximum of N requests in the systems (queued and in service.) We assume that $N > c$, since otherwise there would be no queue build up possible. We also assume that the rate of arrivals and the parameters of the service process may depend on the current number of requests in the system, denoted by n. This type of state dependence is useful, in particular, to represent arrivals from a finite number of exponential sources and service process which varies with the workload. The detailed notation used in our paper is given in Table 1.

We consider the stationary behavior of such a queue. As mentioned in the introduction, the state of our system could be fully described by the total current number of requests in the system and the numbers of requests in each phase of the service process or, alternatively, by the current total number of requests and the current phase of each server. Instead of such a full state description, we propose to use a reduced state description in which we select one server among the c servers and describe the system by the total number of requests and the current phase of the selected server, (n,i). For $n < c$, with probability $(c-n)/c$ the selected server may be idle, in which case we use the value $i = 0$ to denote its idle state.

Let $p(n,i)$ be the steady-state probability corresponding to this reduced state description. The general balance equation for $c < n < N$ is given by

$$p(n,i)[\lambda(n)+\mu_i+v(n,i)] = p(n-1,i)\lambda(n-1)+\sum_{j=1}^{b} p(n,j)\mu_j q_{ji}+\sum_{j=1}^{b} p(n+1,j)\mu_j\hat{q}_j\sigma_i + p(n+1,i)\nu(n+1,i). \quad (1)$$

The corresponding balance equations for other values of n are given in the Appendix.

Note that, for simplicity, we omit in the balance equations the possible dependence on n for the parameters of the service process. In the above equation, $\nu(n,i)$ denotes the conditional rate of departures (request completions) by servers other than the selected server given the current state (n,i). Clearly, we need a way to determine $\nu(n,i)$ for our reduced state equations to be of use.

We denote by $u(n)$ the overall departure rate from the set of c servers given that the current number of requests in the system is n. Using, for example, the second full state description mentioned above, this conditional completions rate can be expressed as

$$u(n) = \sum_{\vec{i}} p(\vec{i} \mid n) \sum_{k=1}^{c} \mu_{i_k}\hat{q}_{i_k} , \quad (2)$$

where $p(\vec{i} \mid n)$ is the conditional probability of the current service phase of each of the c servers given the current number of request in the system. The first sum in formula (2) is over all possible sets of server phases.

The steady-state probability that there are n requests in the system, denoted by $p(n)$, can be expressed as

$$p(n) = \sum_{i=0}^{b} p(n,i) \quad (3)$$

or, alternatively, computed as

$$p(n) = \frac{1}{G}\prod_{k=1}^{n}\frac{\lambda(k-1)}{u(k)}, \qquad n = 0,1,...,N \quad (4)$$

G is a normalizing constant such that $\sum_{n=0}^{N} p(n) = 1$.

Let $\omega(n)$ be the rate of completions for the selected server. Using our reduced state description, we have

$$\omega(n) = \sum_{i=1}^{b} p(n,i)\mu_i\hat{q}_i / p(n) . \quad (5)$$

Since the servers are homogenous, for $n \geq c$ we must have

$$u(n) = c\omega(n) . \quad (6)$$

Recall that, to be able to solve the balance equations, we need the conditional rates of departure $\nu(n,i)$. We use the following intuitive approximation

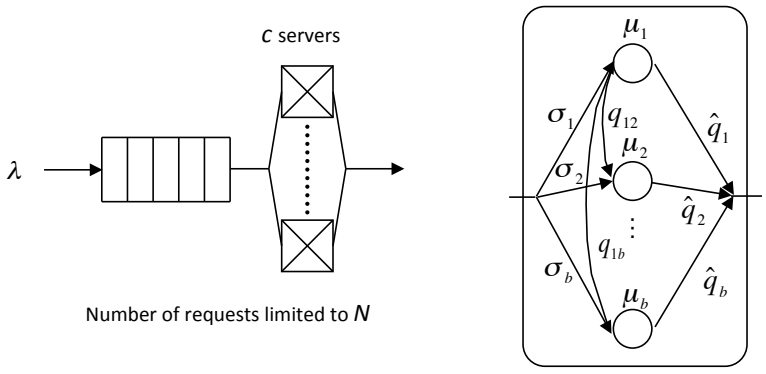
$$\nu(n,i) \approx u(n) - \omega(n) . \quad (7)$$

For $n \geq c$, the above approximation amounts to assuming $\nu(n,i) \approx (c-1)\omega(n)$. Essentially, we assume that the rate of completions for servers other than the selected server exhibits little dependence on the current service phase for the latter.

The corresponding balance equations and details of the computation of $\nu(n,i)$ for other values of n are given in the Appendix.

Thus, together with equations (3), (5), (6) and (7) we get a system of equations for $p(n,i)$ which can be solved in several different ways. In our numerical examples, we use a simple fixed-point iteration.

Clearly, the size of the state-space (n,i) and hence the number of equations to solve is in general far smaller than with either of the two full state descriptions mentioned earlier. Additionally and importantly, it grows only linearly with the number of servers and the number of phases, while the complexity of the full state description grows combinatorially.



The *M/Ph/c/N* queue

The phase distribution with *b* phases for service times

Figure 1 – *M/Ph/c/N* queue without state dependencies

b	Number of phases for the service time distribution
c	Number of servers
N	Buffer space, i.e. maximum of requests in the systems (queued and in service.)
n	Total current number of requests in the system; $n = 0,...,N$
$\lambda(n)$	Rate of requests arrivals given the current number of requests in the system
$\sigma_i(n)$	Probability that service of a request starts in phase i, $i = 1,...,b$ given n
$\mu_i(n)$	Completion rate for phase i of service process
$q_{ji}(n)$	Probability that service process continues in phase j upon completion of phase i, $j,i = 1,...,b$
$\hat{q}_i(n)$	Probability that service process ends (request departs the system) upon completion of phase i, $i = 1,...,b$
m_s	Mean service time of a request
c_s	Coefficient of variation of request service time
s_s	Skewness of a request service time

$p(n,i)$	Probability that there are n requests in the system and the current phase of the service process is i
$p(n)$	Marginal probability that there are n requests in the system
$u(n)$	Overall departure rate from the set of c servers given that the current number of requests in the system is n
$\omega(n)$	Departure rate from the selected server given that the current number of requests in the system is n (when the server is not idle)
$\nu(n,i)$	Departure rate from servers other than the selected server given that the current number of requests in the system is n and the current phase of the service process at the selected server is i

Table 1 – Notation used in this paper

In the next section we study the behavior of the proposed reduced state description in terms of accuracy and computational complexity.

3. NUMERICAL RESULTS

Since performance measures such as the mean number of requests in the system in an *M/Ph/c* queue are known to depend on the shape of the service time distribution (and not only its first two moments) [GUP07, WHI80, WOL77], we organize our exploration as follows. We consider several sets of values for the number of servers c and the maximum number of requests in the system N. We also consider several sets of values for the coefficient of variation c_s of the service time, and we build several distributions with different higher-order properties, viz. skewness.

To build such distributions we keep the mean service time m_s at 1, and we use the algorithm by Bobio *et al.* [BOB05]. The values of skewness s_s explored range from c_s to 100. Recall that the skewness of a random variable S is considered to be a measure of the asymmetry of the underlying distribution and is defined as $s_s = \frac{E[(S-m_s)^3]}{Var[S]^{3/2}}$ where $Var[S]$ denotes the variance of S. Generally, larger value of s_s correspond to longer-tailed distributions. The algorithm used aims to produce a phase-type distribution with the minimum number of phases to match the specified first three moments of the distribution. In our numerical examples, the number of phases varies between 2 and 13 (most frequently around 4).

The performance metrics considered include the mean number of requests in the system (relative error, in Figures 2 and 3), the loss probability (absolute error, in Figure 4) and the overall shape of the steady-state probability distribution $p(n)$ in Figure 5. We define the percentage relative error of our reduced state description versus the actual values as the ratio $100 \times (approximate - actual) / actual$. The actual values are obtained from a numerical solution of the full-description balance equations for the number of servers $c < 128$, and by discrete-event simulation for larger values of c. In the simulation runs we use 7 independent replications with 10,000,000 completions per replication. In all examples in this paper we consider a simple Poisson arrival process with rate λ. Note that with the exception of Figure 5, each figure corresponds to hundreds of data points explored, and the surfaces shown are obtained using an interpolation from sets of scattered data points.

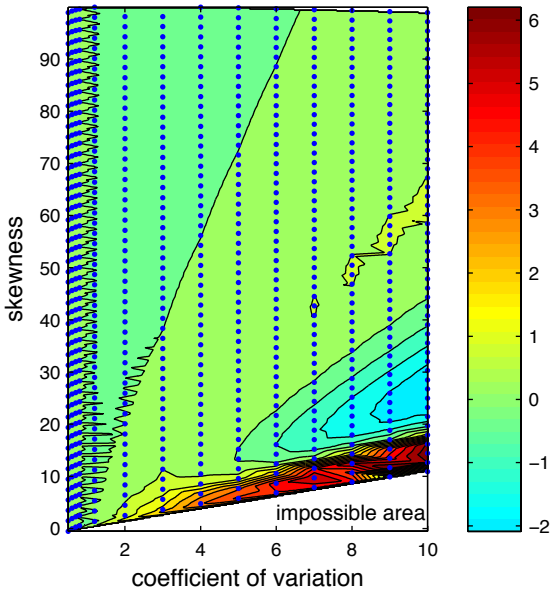


Fig. 2a – Relative errors of the approximate solution for the mean number of requests in *M/Ph/c/N* queue with $c = 8$, $N = c + 10$ and $\lambda = 0.8 \times c$

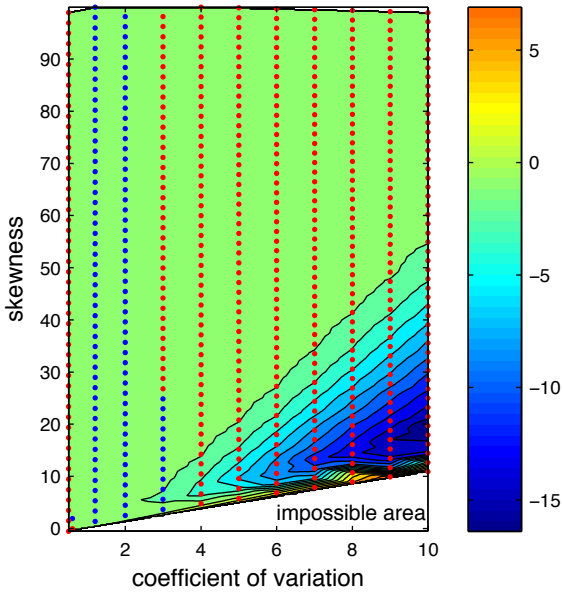


Fig. 2b – Relative errors of the approximate solution for the mean number of requests in *M/Ph/c/N* queue with $c = 64$, $N = 50 + 4 \times c$, and $\lambda = 1.2 \times c$

In Figure 2a, we show the accuracy of the proposed method for an *M/Ph/c/N* queue with $c = 8$, $N = c + 10$ and offered load $\lambda = 0.8 \times c$ (i.e., moderately high utilization) for a range of values of the coefficient of variation c_s and skewness s_s of the service time distribution. We note that the relative error in the mean number of requests in the system is generally small - around 1%. It can reach around 6% for some distributions, which correspond (in this case) to larger values of c_s (say, over 5) and a narrow band of skewness values, viz. small skewness. Note that in a non-negative distribution with a given coefficient of variation c_s the skewness s_s must be greater than a certain value (see Appendix). This is the reason behind the white "impossible area" band in Figure 2.

Figure 2b illustrates the accuracy of our method for a larger number of servers $c = 64$ with a and significantly larger queueing room $N = 50 + 4 \times c$, i.e., $N = 306$, and the offered of $\lambda = 1.2 \times c$. In this set the relative error in the mean number of requests in the system remains under 5% (and most far less) for coefficients of variation c_s not exceeding 5. For larger values of c_s, the relative errors can be larger, attaining 15% for a coefficient of variation of 10 and a small set of values of the skeweness, although, overall, even for such larger values of c_s the relative errors remain well below 5%.

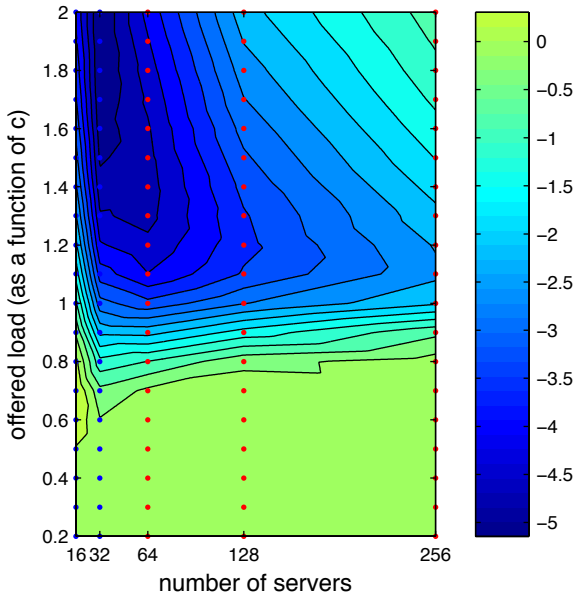


Fig. 3a – Relative errors of the approximate solution for the mean number of requests in *M/Ph/c/N* queue with $N = c + 10$, $c_s = 5$ and $s_s = 6$

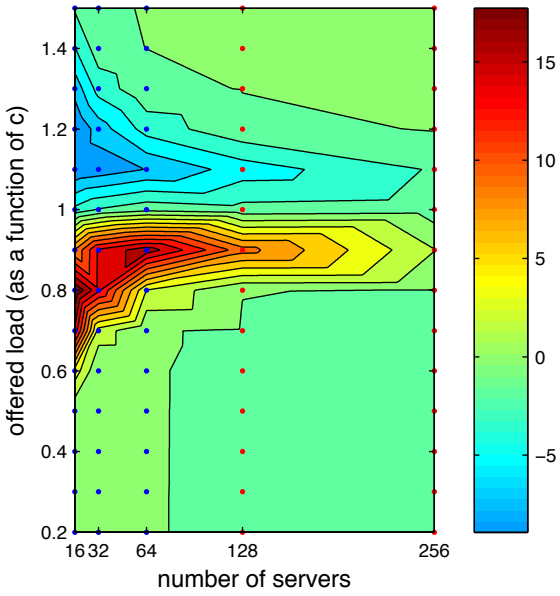


Fig. 3b – Relative errors of the approximate solution for the mean number of requests in *M/Ph/c/N* queue with $N = 4 \times c + 20$, $c_s = 5$ and $s_s = 15$

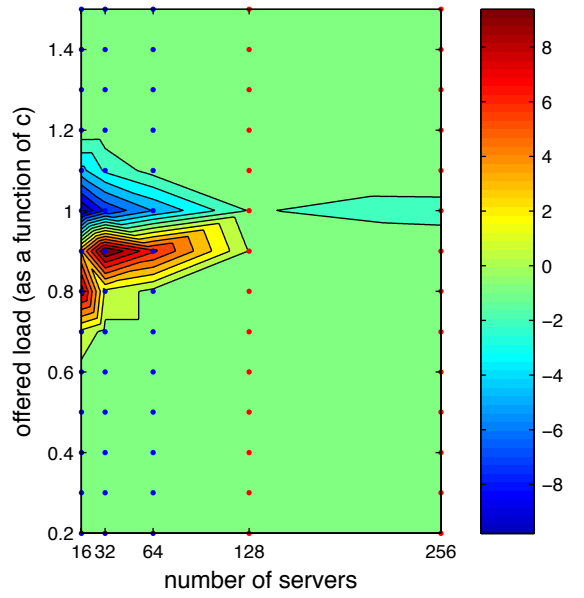


Fig. 3c – Relative errors of the approximate solution for the mean number of requests in *M/Ph/c/N* queue with $N = 4 \times c + 20$ $c_s = 5$ and $s_s = 60$

In Figure 3a, we study the relative error in the mean number of requests in the system as a function of the number of servers and of the offered load for maximum total number in the system $N = c + 10$. The values of the coefficient of variation $c_s = 5$ and of the skewness $s_s = 6$ were selected on purpose since they correspond to a not particularly favorable case, as can be seen

from Figure 2a. We observe that the relative error stays below 4% and clearly tends to decrease as the number of servers increases.

In Figure 3b, we show analogous results for a larger buffer space $N = 4 \times c + 20$. Again, on purpose, we select a service time distribution that corresponds to larger relative errors in Figure 2b, viz. $c_s = 5$ and $s_s = 15$. Here, too, we note that the relative errors for the mean number of requests in the system decrease as the number of servers increases, ranging from around 10% for $c = 32$ to no more than 2% for 256 servers.

Figure 3c illustrates what seems to be typical accuracy of the proposed method for $c_s = 5$. The example shown corresponds to skewness $s_s = 60$ and buffer capacity $N = 4 \times c + 20$. Notice that the relative error for the mean number of requests in the system is below 10% for 16 servers, and becomes virtually negligible when the number of servers exceeds 64.

Examining the reasons for deviations between the exact values and those produced by our approach, we note that the only approximation in our reduced state solution is in the computation of the conditional completion rates of "other" servers given the number of request in the system and the current phase of the selected server $\nu(n,i) \approx u(n) - \omega(n)$ (our formula (7)). It is not surprising then that the deviations tend to decrease and vanish as the number of servers increases. Indeed, the knowledge of the current phase of just one out of many servers does not convey much knowledge about the state of the other servers.

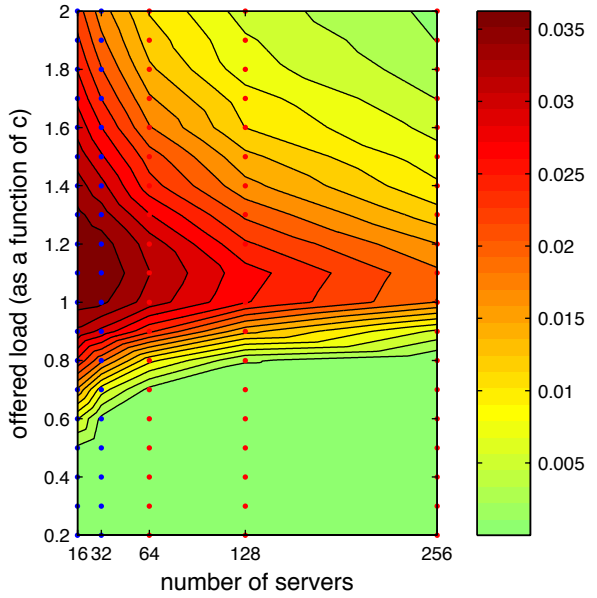


Fig. 4a – Absolute errors for the loss probability in *M/Ph/c/N* queue with $N = c + 10$, $c_s = 5$ and $s_s = 6$

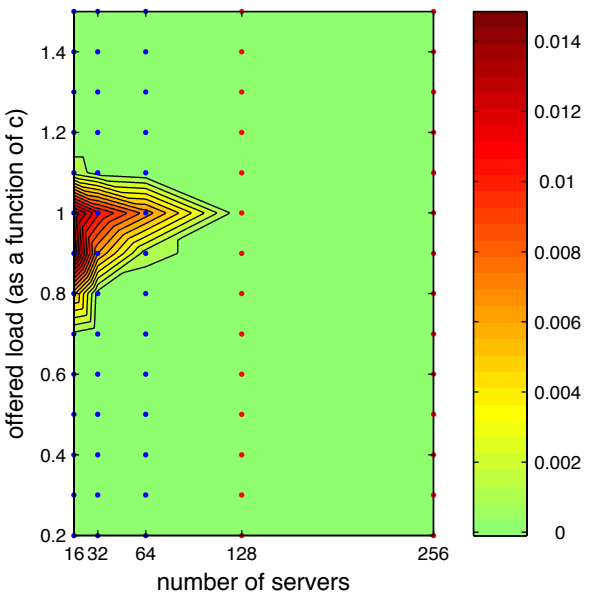


Fig. 4b – Absolute errors for the loss probability in *M/Ph/c/N* queue with $N = 4 \times c + 20$ $c_s = 5$ and $s_s = 60$

Figure 4 shows the absolute deviation between the actual loss probability and that obtained using our reduced state approximation. The results in Figure 4a correspond to queue with a small buffer size of $N = 10 + c$. As in Figure 3a, we selected $c_s = 5$ and $s_s = 6$. We note that the absolute deviation remains below 0.04, and tends to decrease as the number of servers increases (all other things being equal.)

Figure 4b illustrates the deviation in the loss probability in the case of a much larger buffer, i.e., $N = 20 + 4c$. As in Figure 3c, the selected distribution corresponds to $c_s = 5$ and $s_s = 60$. Here, the deviation remains below 0.03 and decreases rapidly with the number of servers.

It is worthwhile mentioning that the deviation in the loss probability is generally much smaller if the coefficient of variation of the service distribution does not exceed, say, 3.

Since our method produces (approximate) values for $p(n)$, the steady-state probability that there are n requests in the system, as an example, we compare in Figure 5 the actual and the approximate values for this probability. Here, we use the same values of the coefficient of variation and skewness as in Figures 3c and 4b for $c = 32$ servers and offered load $\lambda = 0.8 \times c$. We observe that the shape of the $p(n)$ distribution is well reproduced. Clearly, the agreement between actual and approximate values may be less perfect in some cases, but, overall, the shape of the distribution tends to be reproduced correctly.

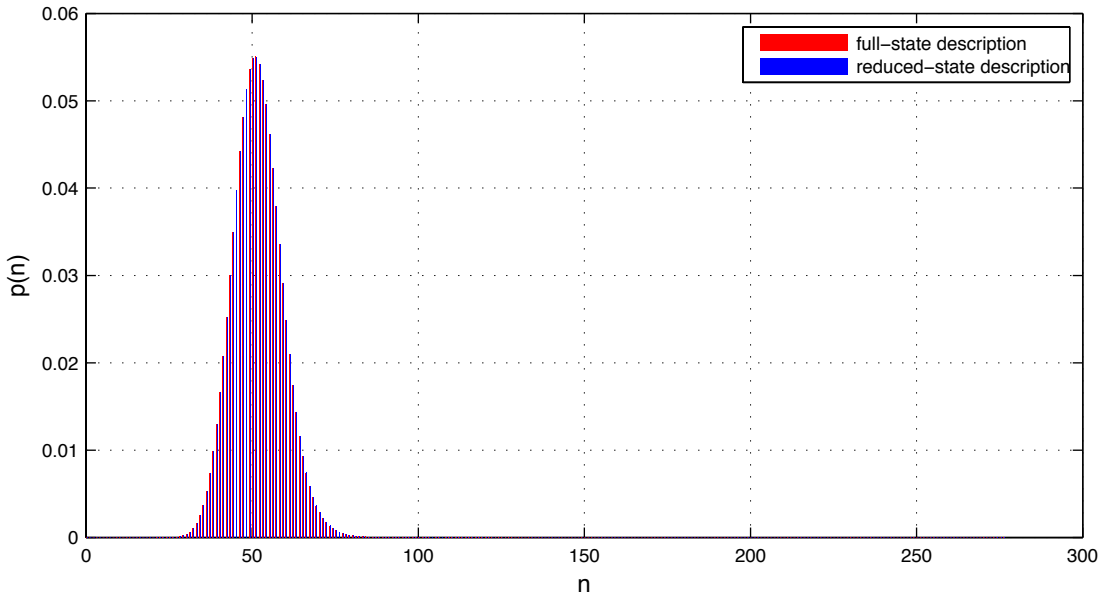


Fig. 5 – Comparison between actual and approximate values for steady-state probability $p(n)$ in *M/Ph/c/N* queue with $c = 32$, $\lambda = 0.8 \times c$, $N = 4 \times c + 20$, $c_s = 5$ and $s_s = 60$

We now consider the issue of computational complexity of the proposed approach. As mentioned in the introduction, there are essentially two possible full state descriptions for an *M/Ph/c* queue. The first description consists of the number of requests and the vector of the current number of servers in each phase of the service process. The total number of states for this description is given by $1 + \sum_{n=1}^{c-1} \binom{b+n-1}{n} + (N-c+1)\binom{b+c-1}{c}$. The second description involves the current number of requests in the system and the vector of the current phases for each server. The total number of states in this state description is given by $1 + \sum_{n=1}^{c-1} \binom{c}{n} b^n + (N-c+1)b^c$. The number of states in the proposed reduced state description is given by $1 + (c-1)(b+1) + (N-c+1)b$.

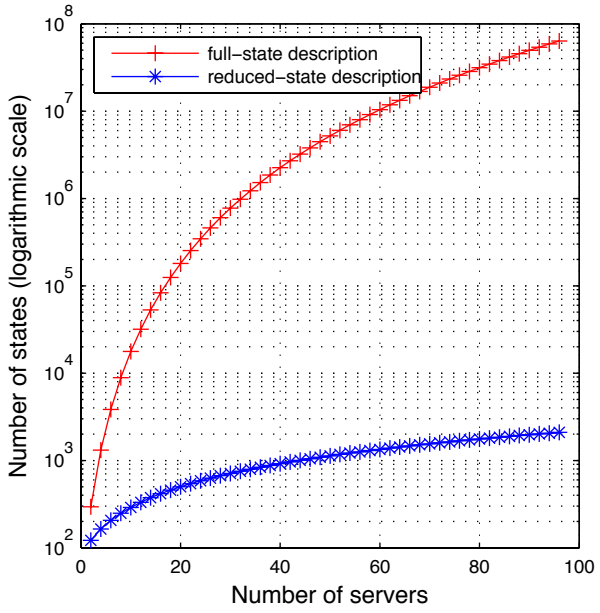


Fig.6a – Comparison between the number of states in the full and the reduced state description for *M/Ph/c/N* queue with $b = 4$ and $N = 4 \times c + 20$

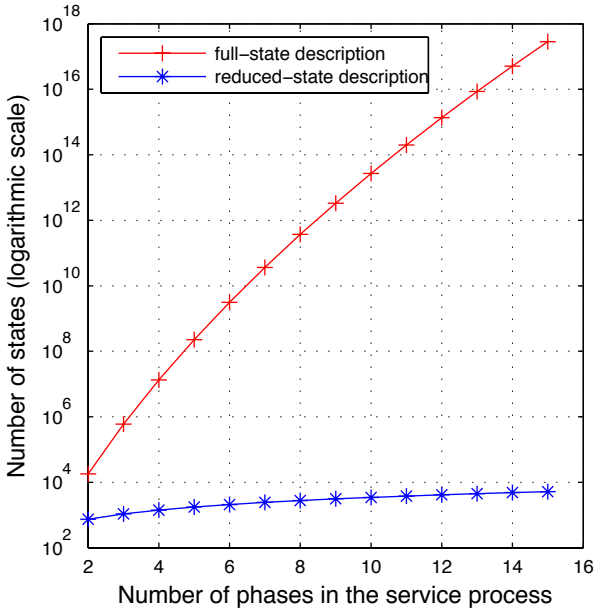


Fig.6b – Comparison between the number of states in the full and the reduced-state description for *M/Ph/c/N* queue with $c = 64$ and $N = 4 \times c + 20$

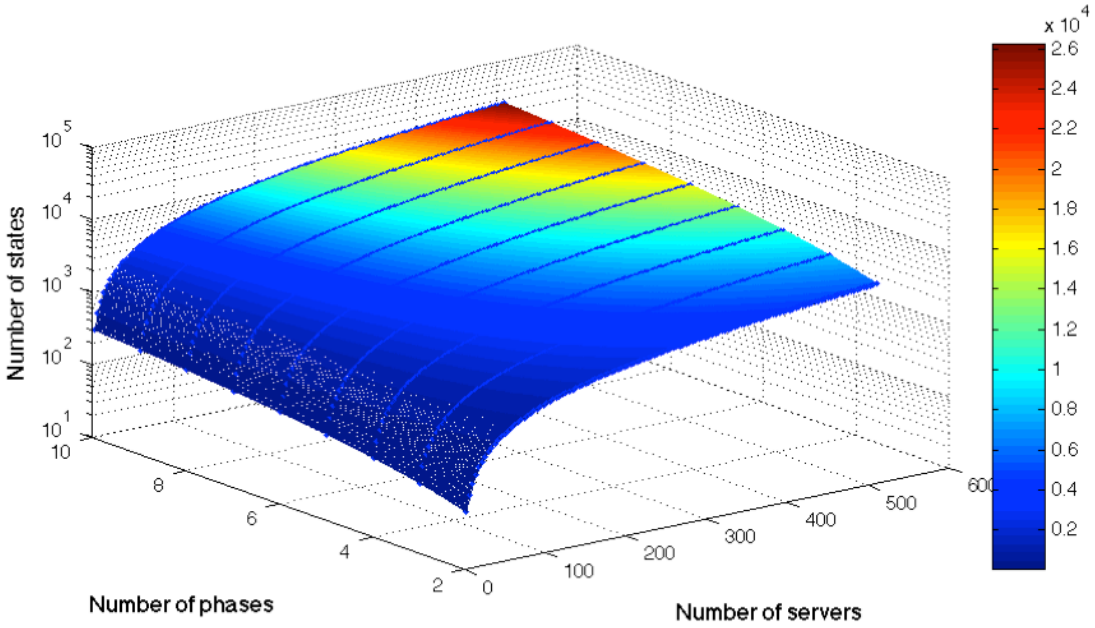


Fig.6c – Evolution of the number of states in our reduced-state description as a function of b and c

Figure 6 compares the computational complexity in the number of states between the first full description and our reduced state description. We do not include the second full state description in our figure since the number of states is systematically higher than for the first one. It is obvious from Figure 6 that, as the number of servers and phases increases, there is a difference of many orders of magnitude between the complexity of the full state description and our proposed reduced state description. Even with a relatively small number of phases (say 4), the complexity of the full state description results in about 13,000,000 states while the reduced state description involves less than 1,500 states with only 64 servers (see Figure 6 a). The

difference gets even more dramatic for larger numbers of phases. With 8 phases and 64 servers, there are over 3 trillion states in the full description compared with less than 3200 states for our method, amounting to 9 orders of magnitude difference (see Figure 6 b).

Figure 6c shows the number of states in our reduced state description for a wide range of values of the number of servers and the number of phases. The gradual linear increase in the complexity is self evident.

The next section presents the conclusions of this paper.

4. CONCLUSION

This paper presents an approach, believed to be novel, to the solution of a queueing system with multiple homogenous servers, quasi-general service times (phase-type distributions), quasi-Poisson arrivals and a limited buffer space (queueing room.) The proposed approach uses a reduced state description in which the state of only one server is represented explicitly while the other servers are accounted for through their rate of completions.

The resulting accuracy is generally good and, importantly, tends to improve as the number servers in the system increases. This conclusion is supported by a large number of data points only a small fraction of which is shown in this paper.

In the classical state description used until now for this type of queueing system, the number of states grows combinatorially, making the problem intractable for larger numbers of servers and / or phases. By contrast, the computational complexity in terms of the number of states in our reduced state description grows only linearly in the number of servers and phases. This, for the first time, puts problems with hundreds of servers and several phases within easy reach of a fast numerical solution.

Future work includes extension of our approach to the case of unrestricted queueing room, as well as quasi-general times between request arrivals.

REFERENCES

[BOB05] Bobbio, A., Horvath, A., and Telek, M., Matching three moments with minimal acyclic phase type distributions, *Stochastic Models*, Vol. 21, 2005, pp. 303-326.

[GOU96] Gouweleeuw, F.N., and Tijms, H., A simple heuristic for buffer design in finite-capacity queues. *European Journal of Operational Research*, Vol. 88 (3), 1996, pp. 592-598.

[GUP07] Gupta, V., Harchol-Balter, M., Dai, J. and Zwart, B., The effect of higher moments of job size distribution on the performance of an *M/G/s* queueing system, *Performance Evaluation Review*, Vol. 35 (2), 2007, pp. 12-14.

[HOK78] Hokstad, P., Approximations for the *M/G/m* Queue, *Operations Research*, Vol. 26 (3), 1978, pp. 510-523.

[JOH88] Johnson, M. A., and Taaffe, M. R, The denseness of phase distributions. *School of Industrial Engineering*, Purdue University. 1988.

[KIM93] Kimura, T., Equivalence relations in the approximations for the *M/G/s/s+r* queue, *Mathematical and computer modeling*, Vol. 31 (10), 2000, pp. 215-224.

[KIM96] Kimura, T., A transform-free approximation for the finite capacity *M/G/s* queue, *Operations Research*, Vol. 44 (6), 1996, pp. 984-988.

[LAT93] Latouche, G. and Ramaswami, V, A logarithmic reduction algorithm for quasi-birth-and-death processes, *Journal of Applied Probability*. Vol. 30, 1993, pp. 650-674.

[LAT94] Latouche, G., Newton's iteration for non-linear equations in Markov chains, *IMA Journal of Numerical Analysis*, Vol. 14,

1994, pp. 583-598.

[MIY86] Miyazawa, M., Approximation of the Queue-Length Distribution of an *M/GI/s* Queue by the Basic Equations, *Journal of Applied Probability*, Vol. 23 (2), 1986, pp. 443-458.

[RAM85a] Ramaswami, V., and Lucantoni, D. M., Stationary waiting time distribution in queues with phase type service and in quasi-birth-and-death-processes, *Stochastic Models*, Vol. 1, 1985, pp. 125-136.

[RAM85b] Ramaswami, V., and Lucantoni, D. M., Algorithms for the multi-server queue with phase type service, *Stochastic Models,* Vol. 1, 1985, pp. 393-417.

[SEE86] Seelen, L. P., An Algorithm for *Ph/Ph/c* Queues, *European Journal of the Operations Research Society*, Vol. 23, 1986, pp. 118-127.

[SMI03] Smith, J. M., *M/ G/c/K* blocking probability models and system performance, *Performance Evaluation*, Vol. 52 (4), 2003, pp. 237–267.

[WHI80] Whitt, W., The effect of variability in the *GI/G/s* queue, *Journal of Applied Probability*, Vol. 17 (4), 1980, pp. 1062-1071.

[WOL77] Wolff, R.W., The Effect of Service Time Regularity on System Performance, *Computer Performance*, North Holland, 1977, pp. 297-304.

APPENDIX

A.1 Balance equations

We give here the balance equations for the cases not explicitly treated in the body of the paper, starting with the case $0 < n < c$.

$$p(n,i)[\lambda(n)+\mu_i+v(n,i)] = p(n-1,0)\lambda(n-1)\sigma_i/(c-n+1) + p(n-1,i)\lambda(n-1)*(c-n)/(c-n+1) + \sum_{j=1}^{b} p(n,j)\mu_j q_{ji} + p(n+1,i)\nu(n+1,i), \qquad i=1,...,b$$

$$p(n,0)[\lambda(n)+v(n,0)] = \sum_{i=1}^{b} p(n+1,i)\mu_i\hat{q}_i + p(n+1,0)\nu(n+1,0) + p(n-1,0)\lambda(n-1)(c-n)/(c-n+1) \quad .$$

Note that we use the notation $p(n,0)$ to denote the case when the selected server is idle, which, in our model, can only happen if $n < c$. Note also that we have $\nu(n=1, i \neq 0) = 0$.

In the case $n = 0$, we can only have

$$p(0,0)[\lambda(0)] = \sum_{i=1}^{b} p(1,i)\mu_i\hat{q}_i + p(1,0)\nu(1,0) \; .$$

For $n = c$ we have

$$p(c,i)[\lambda(c)+\mu_i+v(c,i)] = p(c-1,0)\lambda(c-1)\sigma_i + p(c-1,i)\lambda(c-1) + \sum_{j=1}^{b} p(c,j)\mu_j q_{ji} + p(c+1,i)v(c+1,i) + \sum_{j=1}^{b} p(c+1,j)\mu_j\hat{q}_j\sigma_i, \qquad i=1,...,b \quad .$$

Finally, for $n = N$ we obtain

$$p(N,i)[\mu_i + \nu(N,i)] = p(N-1,i)\lambda(N-1) + \sum_{j=1}^{i-1} p(N,j)\mu_j q_{ji}, \qquad i = 1,...,b .$$

The conditional rate of completions for the selected server when it is not idle and there are n requests in the system is given by $\omega(n) = \sum_{i=1}^{b} p(n,i)\mu_i \hat{q}_i / \sum_{j=1}^{b} p(n,j)$, and can also be expressed as $\omega(n) = u(n)/\min(n,c)$. As before, we approximate the conditional rates of departure $\nu(n,i)$ for $i = 1,,,,b$ by $\nu(n,i) \approx u(n) - \omega(n)$. For $0 < n < c$, we use $\nu(n,0) \approx u(n)$.

A.2 Constraints on skewness

Let Z be a non-negative random variable, and denote by m_i its i-th moment $E[Z^i]$ and by n_i its i-th normalized moment. We have $n_2 = m_2 / m_i^2$ and $n_3 = m_3/(m_1 m_2)$. We must have (see [Osogami]) $n_3 \geq n_2$. It follows the skewness of Z, denoted by S_Z, must satisfy the relationship

$$s_z \geq m_1(c_Z - 1/c_Z) ,$$

where $c_Z = (m_2/m_1^2 - 1)^{1/2}$ and $s_Z = (m_3 - 3m_1 m_2 + 2m_1^3)/(m_2 - m_1^2)^{3/2}$. c_Z is the coefficient of variation of the random variable Z. In our case, $m_1 = 1$ so that we must have $s_s \geq (c_s - 1/c_s)$.

RESEARCH CENTRE
GRENOBLE - RHÔNE-ALPES

Inovallée
655 avenue de l'Europe - Montbonnot
38334 Saint Ismier Cedex France

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr
ISSN 0249-6399